An MBSE-based Approach to Architecting a Robotic Sample Capture System Concept for Potential Mars Sample Return

Paulo Younse, Jessica Cameron
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109 USA
paulo.j.younse@jpl.nasa.gov

***Abstract*—A model-based systems engineering (MBSE) approach was applied to architecting an orbiting sample Capture and Orient Module (COM) system concept for a Capture, Contain, and Return System (CCRS) payload concept for the notional Mars Sample Return (MSR) campaign at the NASA Jet Propulsion Laboratory. An architecture framework was established, covering multiple organizational layers of the system, along with structural, behavioral, data, and requirements perspectives. A workflow process to implement the architecting activities within the COM engineering team was established. The approach helped maintain consistency in terminology, helped ensure alignment of structural, behavioral, data, and requirements elements within each organization layer, and guided the engineering team through an architecting process that helped develop the architecture for a Capture and Orient Module system concept.**

TABLE OF CONTENTS

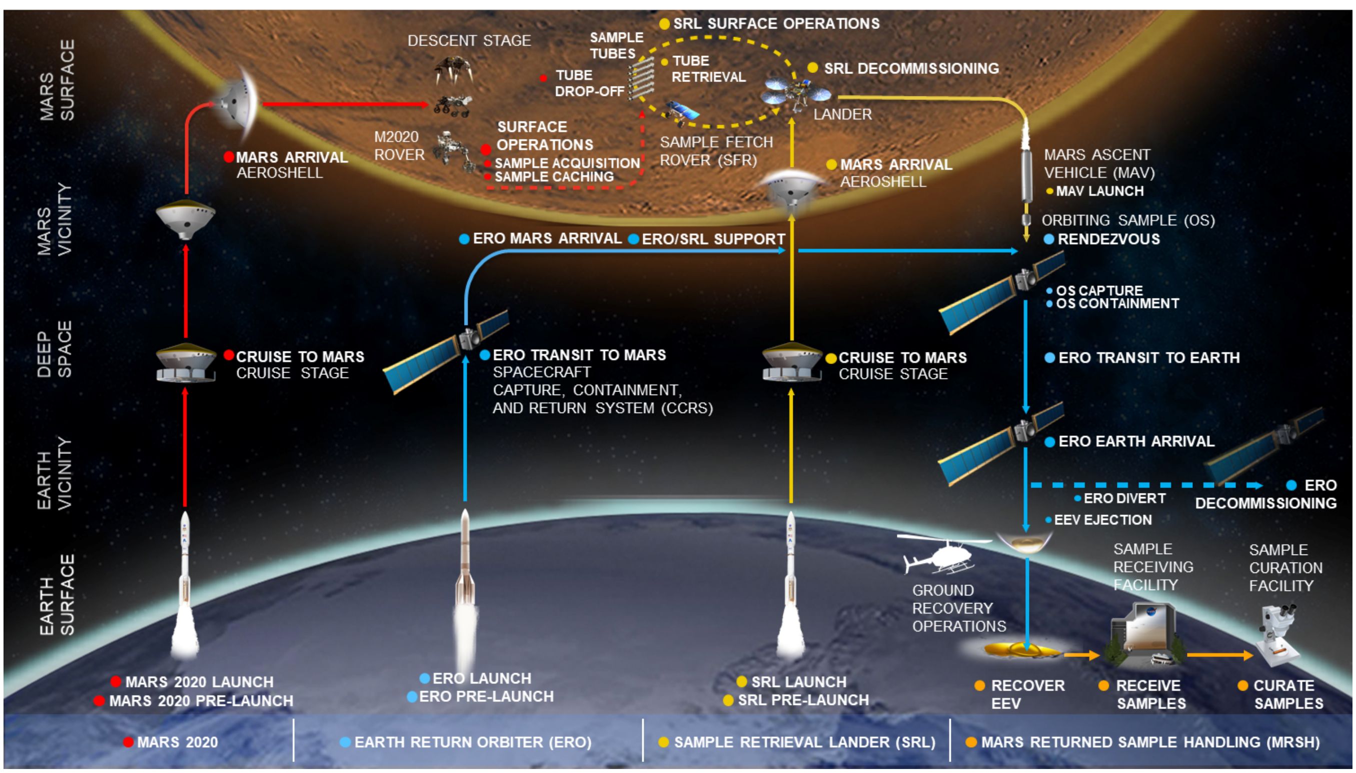

Figure 1. Notional MSR architecture. Note that all elements beyond Mars 2020 are conceptual [1] (Credit: O. Rehman).

1. INTRODUCTION

On August 5, 2012, NASA's 900 kg Mars Science Laboratory Curiosity rover successfully landed on the surface of Mars and set out to search for evidence of past habitable environments [2] [3]. The Curiosity rover pushed the boundaries of technology and systems engineering, consisting of approximately 50,000 parts, involving nearly 3,000 NASA employees and 4,000 non-government workers, and was considered the most complex rover of its time ever sent to another planet [2] [4] [5].

Despite the technical and scientific achievements of the rover, the project experienced numerous development challenges, and in the end, saw an increase in over $881 million in costs from its original 2008 project baseline, as well as a 26-month launch delay due to design and technical problems that necessitated late design changes in hardware, avionics, and software [6]. A metric for design changes used by NASA is "drawing growth" after the Critical Design Review (CDR), with which MSL saw a 147% growth [7]. Some of these late design changes were attributed to the discovery of divergent requirements discovered late during the testing phase. These divergent requirements were found to be a consequence of not having an architecture to pull together and manage in a cohesive manner the complex web of documentation of system and subsystem functional requirements, environmental requirements, interface control documents, institutional policy documents, and planetary protection requirements [8].

In summer of 2020, NASA is planning to launch its next rover to Mars, Mars 2020, with the goal to search for evidence of past life on Mars, as well as collect a set of samples for potential return to Earth [9] [10]. Mars 2020 reuses roughly 85% of the engineering design of MSL, but will carry new hardware and instruments for sample collection, and to search for biosignatures [11] [12]. Mars 2020 is currently facing similar technical problems and design challenges as MSL did, also requiring design changes and resulting in cost growth and schedule delays [13]. For both of these robotic space system project examples, late design changes were a major factor in cost growth and schedule delays. In fact, NASA has observed that major projects that have rebaselined cost and schedule tended to have experienced more of these late design changes [14].

In general, space missions are considered high-risk systems and more prone to accidents due to their tightly coupled systems and need to manage complex interactions [15]. Due to the multidisciplinary nature of space missions, the complexity of the problem extends to complexity in requirements, design, flight software development, testing, and operations, which has been correlated to higher spacecraft cost and lower rates of mission success [15]. Robotic spacecraft systems, in particular, are becoming increasingly more complex, and hardware capability and software complexity is expected to continue to grow [16]. Using lines of code are one indicator of complexity, history shows an exponential growth trend in flight software complexity for robotic missions [15]. If this trend in complexity growth continues, higher spacecraft costs and mission success risk for future robotic space missions can also be expected to grow.

NASA's Jet Propulsion Laboratory (JPL) stated through its 2018 Strategic Implementation Plan that it will "pursue our long-term scientific Quests with a diverse and bold portfolio of missions as we push the limits of space exploration technology by developing and fielding ever more capable autonomous robotic systems" [17]. A potential set of future missions under study by NASA and ESA that would push these limits are proposed for the Mars Sample Return (MSR) campaign (see Fig. 1). The current notional MSR campaign architecture consists of two follow-up missions to Mars 2020, consisting of a Sample Return Lander (SRL) mission, and an Earth Return Orbiter (ERO) mission. SRL would land on Mars with a fetch rover to retrieve the Mars 2020 samples and place them into Mars orbit within an Orbiting Sample (OS) container. ERO would robotically capture the OS and return it to Earth within an Earth Entry Vehicle (EEV) with a Capture, Contain, and Return System (CCRS) [18] [19] [20]. If NASA and JPL are to succeed in future robotic space missions like those associated with Mars Sample Return, management of the growth in complexity associated with these future missions will be critical in order to control costs, maintain schedule, and ensure mission success.

Model-based Systems Engineering (MBSE) provides techniques to aid in the development of these types of complex systems by aiming to reduce both design errors and rework, as well as improving system quality and project performance over traditional systems engineering techniques [21] [22]. MBSE seeks to improve on the state of the art in systems engineering through enhancing communications to aid in system presentation and understanding, reducing development risk through enabling ongoing requirements V&V and more accurate cost estimation, improving system quality in terms of requirements and requirements traceability, increasing productivity with systems engineering activities, and enhancing knowledge transfer through more effective capturing of domain knowledge in a standardized form [23]. Additionally, MBSE places a great focus on systems engineering activities at the earlier stages of the project life cycle, which aim to reduce the risk of the accruing magnified costs associated with dealing with defects detected later in the project life cycle that could trigger design changes. The aerospace industry in particular has high potential for overall project cost reductions by moving from traditional systems engineering to MBSE-based on the industry's tendency to possess projects with high system complexity, high environment complexity, and long lifespans, where these benefits would be more evident and enhanced relative to other industries [24]. Within the aerospace industry, robotic space systems feature close integration of large numbers of subsystems with highly integrated, complex, and intelligent structural and behavioral autonomous elements from multiple domains such as mechanical, electrical, control, and software [25] [26]. A MBSE approach can help engineers deal with the complexity of these robotic systems, as well as perform the thorough analysis at the foundational systems level necessary to realize these systems [25] [26].

Despite the proposed benefits of MBSE, its practice has yet to be widely adopted [27]. Some causes for low MBSE adoption include: technical issues, cultural issues, and economic barriers [28]. Additional challenges described by practitioners from experience applying MBSE on recent JPL flight projects include burdens associated with simultaneous introduction of a new MBSE approach, tool development, and processes, as well as negative impacts of project work due to immature tools and lack of understanding of user workflow [29]. The SysML language, which is often used within an MBSE approach, also has not gained widespread adoption of popularity in robotics, though research has presented SysML as a viable framework in which to model robotic systems [30]. There currently is a lack of substantial and compelling evidence in the literature to promote broad adoption of MBSE for robotic space systems. Robotic space systems have much to benefit from an MBSE approach due to their intrinsic complexity, particularly if implemented during the early phases of the project such as system architecting.

The purpose of this research is to investigate the benefits of MBSE for architecting robotic space systems. The notional MSR campaign's ERO mission concept CCRS Capture and Orient Module (COM) system is used as a case study to demonstrate these benefits.

2. BACKGROUND

The Capture and Orient Module (COM), shown Fig. 2, performs the initial operations of the Capture, Contain, and Return Systems (CCRS). The functions of the COM are to capture, constrain, orient, inspect, and assemble the Orbiting Sample (OS) into the Primary Containment Vessel (PCV) in preparation for sealing and installation into the Earth Entry Vehicle (EEV). The EEV would eventually deliver the OS back to Earth. Various architectures for OS capture systems have been studied and proposed over the past 20 years [31] [32] [33] [34] [35]. The current concept for the COM is shown in Fig. 3. The COM consists of the following functional elements:

- Capture Mechanism to contain any unsterilized Martian dust arriving with the OS on its surface
- Sensor System to trigger closure the Capture Mechanism and Transfer Mechanism during OS capture and inspect the surface of the OS
- Capture Cone to catch and contain the OS
- Orientation Mechanism to constrain and orient the OS
- Transfer Mechanism to cage the OS, transfer the OS through the COM subsystems, assemble the OS into the PCV, and maintain preload on the PCV Lid during sealing
- COM Infrastructure to integrate the COM elements on the Capture and Containment Module (CCM)

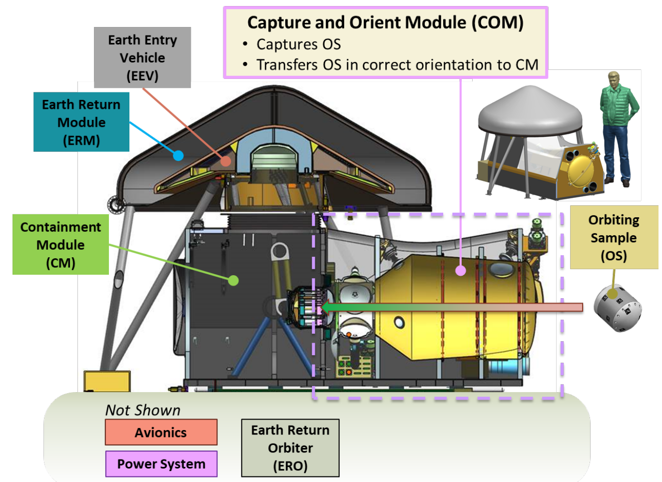

Figure 2. Notional Capture, Contain, and Return System concept [1].

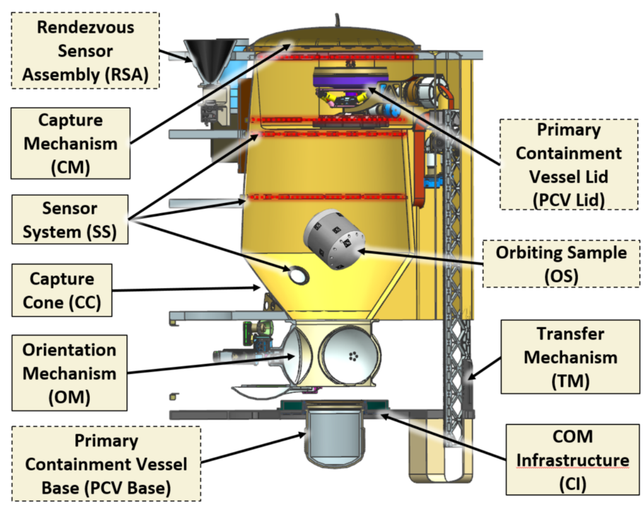

Figure 3. Notional Capture and Orient Module concept [1].

The ERO mission is a robotic mission, it would be operated from a mission control center on Earth. The primary COM operations take place in Mars orbit. Certain operations of the COM, such as OS capture, rely on event-driven sequences that are impossible or impractical to initiate from mission control due to the communication delay that occurs with data transmission between Earth and Mars. Therefore, a level of autonomy would be required for the COM to achieve its goals while operating independent of ground control. This autonomy can be in the form of pre-planned sets of instructions transmitted to the spacecraft and executed by the CCRS Command and Data Handling (C&DH) system [36].

Architecture Scope

The type of architecture considered for the Capture and Orient Module is that of a robotic system architecture, as it relies on integrated mechanical, electrical, and software systems to perform its operations with a level of autonomy. The Capture and Orient Module architecture was classified along four architecture taxonomic dimensions defined in [37]:

- Abstraction: Progression of the architecture from more abstract to more concrete, ranging from the system context to the physical implementation
- Organization: Level of decomposition of the architecture, ranging from an enterprise or system-of-systems down to its individual components
- Categorization: Architectural categories that may possess unique system characteristics, follow particular business processes, serve specific operational uses, or meet particular stakeholder requirements
- Time: Period in the system life cycle during which the architecture is defined and has particular rules associated with its architectural evolution

The architecture space for three of the architecture taxonomic dimensions that was used to classify the COM architecture is shown in Fig. 4. The axis of organization spans from the top-level Mars Sample Return Campaign, within which the system participates, down to the individual system components. The axis of categories were defined as key functional domains relevant to the CCRS project and key to its architecture. The COM architecture scope addresses the module level down to its components, spans through operational to physical definition, and covers robotics, mechanical, electrical, thermal, flight software, contamination control, and planetary protection functional domains.

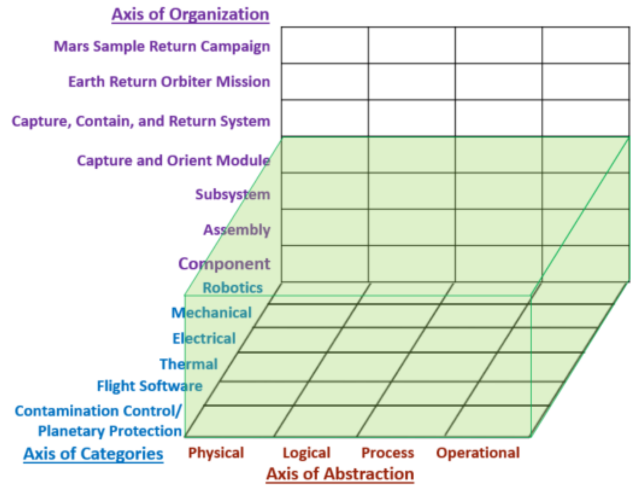

Figure 4. Capture and Orient Module architecture classification shown in green along the axes of abstraction, organization, and categorization based on the architecture taxonomy described [37].

The COM architecture time dimension was defined by the seven phases of the NASA Project Life-Cycle (see Fig. 5). The architecture for an initial, feasible concept of the COM takes place during Pre-Phase A. In Pre-Phase A, the project looks at a range of ideas and alternative architectures, determines the feasibility of the desired system, and develops candidate system concepts [38].

NASA Project Life-Cycle Phases
Pre-Phase A: Concept Studies
Phase A: Concept & Technology Development
Phase B: Preliminary Design & Technology Completion
Phase C: Final Design & Fabrication
Phase D: System Assembly, Integration & Test, Launch and Checkout
Phase E: Operations & Sustainment
Phase F: Closeout

Figure 5. Capture and Orient Module architecture phase boxed in green within the NASA Project Life-Cycle based on NASA Procedural Requirements NPR 7123.1B [39].

3. ARCHITECTING APPROACH

One of the challenges observed with introducing MBSE into a flight project at JPL was the difficulty encountered by team members to simultaneously learn new approaches, tools, and processes associated with the practice while still carrying out their daily engineering activities [29]. Because the MBSE architecting approach developed for the COM was new to the engineering team, overwhelming the team with simultaneous introduction of new architecting processes, tools, and a modeling languages was a legitimate concern. To address this concern, the architecting approach developed separated out the framework to capture the information content for the system architecture, process to develop the architecture, and method to implement the architecture process. This allows the opportunity for aspects of the architecture approach to be incrementally introduced and gradually infused into a team workflow. In particular, isolating the method from the framework and process allows for flexibility to choose to implement the process with either MBSE or non-MBSE (e.g., document-centric engineering, domain-specific design engineering) methods, using specialized tools (e.g., Cameo, Excel, PowerPoint, CAD), and communicated with a preferred language (e.g., SysML, natural language).

Framework

A framework was developed to capture the architecture information content, guide the architecting process, provide views of the system from different perspectives, and arrange the architecture information in a manner that complements MBSE methods (see Fig. 6). The framework was derived from frameworks defined in the MBSAP and MagicGrid approaches described in [37] and [40]. A table format, similar to that used by MagicGrid, was adopted due to its visual representation, which aides in communicating, understanding, and tracking the architecture process. The axis of organization from the system architecture taxonomy shown in Fig. 6 was chosen for the vertical axis of the table. The structure, data, behaviour, and requirements perspectives from the MBSAP approach were chosen for the horizontal axis of the table.

		Perspective			
		Structure	Data	Behavior	Requirements
Organization	Module Level (L4)				
	Subsystem Level (L5)				
	Assembly Level (L6)				
	Component Level (L7)				

Figure 6. Architecture framework used for developing the COM architecture.

Process

The architecting process for developing the COM architecture is shown in Fig. 7. It follows the general flow of developing an Operational Viewpoint, Logical/Functional Viewpoint, and Physical Viewpoint described in [37], as well as the NASA System Design Process described in [38]. Note that even though the processes are shown to proceed in order from 1 through 16, the structure, data, and behavioral aspects for the process are tightly coupled, and their development can evolve together at each organizational level.

		Perspective			
		Structure	Data	Behavior	Requirements
Organization	Module Level (L4)	1	2	3	4
	Subsystem Level (L5)	5	6	7	8
	Assembly Level (L6)	9	10	11	12
	Component Level (L7)	13	14	15	16

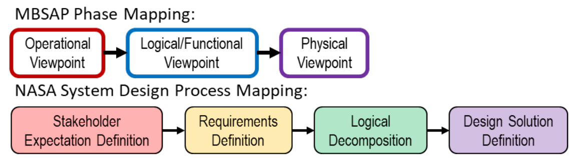

Figure 7. Architecting process implemented to develop the COM architecture, with mapping of the MBSAP Phases and NASA System Design Processes, and numbered by order of operation.

Method

An MBSE method was used for the module, subsystem, and assembly levels of the architecture, while non-MBSE methods were used for the component level (see Fig. 8). For the MBSE method, Cameo Systems Modeler was chosen as the architecting software tool, and SysML was chosen as the architecting language. Cameo was chosen due to its institutional support at JPL, flexibility with processes and viewpoints, availability of technical support and documentation, strict enforcement of OMG SysML syntax and semantics, and simulation capability. SysML was chosen due to the availability of documentation, wide model coverage of systems engineering concepts, visual representation, and familiarity within the systems engineering community. The MBSAP method was applied for modelling the system structure, data, behaviour, and requirements in SysML within Cameo Systems Modeler.

		Perspective			
		Structure	Data	Behavior	Requirements
Organization	Module Level (L4)				
	Subsystem Level (L5)	Applied MBSE Methods			
	Assembly Level (L6)				
	Component Level (L7)	Applied Non-MBSE Methods			

Figure 8. Mapping of MBSE and non-MBSE methods used within the COM architecture framework.

At the component level, Computer Aided Drafting (CAD) tools, Excel, and PowerPoint were used due to their familiarity within the broader engineering team, domain-specific design features, and effectiveness in capturing and communicating more detailed numerical, textual, and graphic design information.

Scope of Architecting Activity

It was necessary to scope the architecting efforts performed by the COM engineering team to fit within the available project resource constraints (i.e. workforce availability, schedule), to enable full completion of the architecting process in order to converge upon at least one full system concept, and operate in an early project environment where many system elements are still ill-defined. Therefore, the architecting activity was limited to:

- Single architecture for a candidate system concept (trades and alternatives at subsystem levels were still captured, and effort was made to develop an architecture that could be flexible and compatible with alternative technologies)

- Primarily the hardware aspects of the robotic system (defining the software and avionics aspects of the architecture are planned for future work)

- Interaction (type B) scenarios that focus primarily on the direct interactions between the system of interest and the actors and external systems within the system context for the top-level COM scenarios [41]

- Main scenarios that focus on the most common sequence of interactions for completing the required operations (development of alternative, exception, and fault scenarios is planned for future work)

- Primary portion of the COM scenario that starts from OS capture and ends with OS assembly into the PCV (development of commissioning, decommissioning, and secondary activities is planned for future work)

4. SYSTEM MODEL

Model Organization

The presentation of the information captured within the SysML COM model should be unproblematic to navigate in order to allow for maximum productivity when updating, referencing, and studying the COM model. The organization of the COM model is captured within a containment tree in SysML (see Fig. 9). The containment tree depicts the system organization as it breaks down to individual diagrams, tables, blocks, requirements, and behaviors. The system model framework outlined by MagicGrid focuses on four basic perspectives: requirements, behaviour, structure, and parametrics. Based on the scope of work the COM system requires at this stage of research and development, the basic organization of the SysML model was modified to better capture the information that is critical to the engineering team.

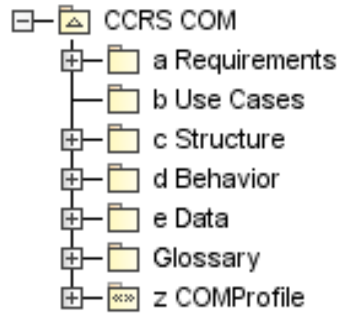

Figure 9. Model containment tree with Cameo Systems Modeler showing model organization.

The CCRS COM model element contains all of the content packages. Within each of these packages, they are further organized into diagrams and sub-packages (see Fig. 10). The packages reflect the MagicGrid pillars, as well as content heavily studied by the engineering team:

- Requirements package contains all requirement diagrams (hierarchy, structure decomposition) as well as the requirements table. Requirements are captured at every level of the model.

- Use Cases package is not currently utilized, but in future will contain use case diagrams that will capture data and power profiles for various mission scenarios.

- Structure package contains a high-level block definition diagram (BDD) of the COM system. The overall BDD contains blocks that describe subsystems. Each block captures an internal block diagram (IBD) that captures and describes the inner workings of the subsystem.

- Behavior package contains the activity diagrams for each high-level COM objective, as well as the lower-level activity diagrams.

- Data package currently contains a block diagram of the data package breakdown. This describes the various types of data the system will use or generate.

- Glossary package houses all of the various acronyms the team uses to abbreviate the system

- COMProfile package houses all of the system stereotypes, signals, as well as a package diagram that describes the system as a whole

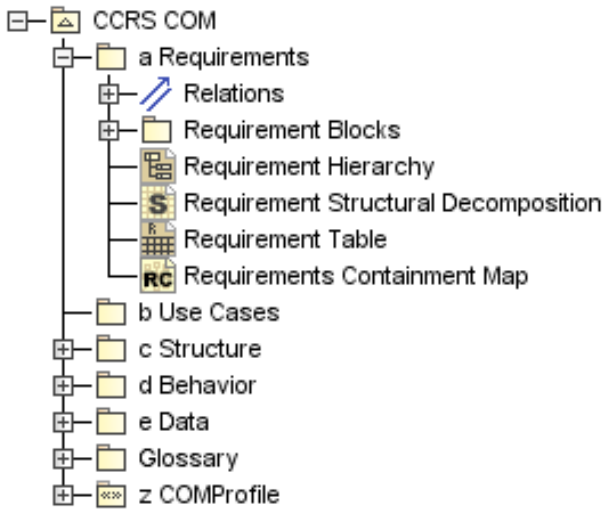

Figure 10. Requirements Package breakdown with sub-packages and diagrams.

Model Overview

It is necessary to construct a diagram within SysML that will provide the engineering team with an overview of the modelling work within the MBSE software. One of the many useful aspects of MBSE is the ability to display system information in different ways than other modelling methods. SysML provides the engineering team with the ability to trace specific relationships between the various modelled components and the overall system. The COM model overview diagram displays the different perspectives (structure, behaviour, requirements, data) for each of the organizational levels modelled (see Fig. 11).

One of the primary goals of the COM system is to position the OS into an acceptable orientation for return to Earth. To accomplish this, an Orientation Mechanism is located at the base of the COM. The OS must travel from the COM entry point (located on the capture plane) to the Orientation Mechanism on the opposite end of the COM. A Transfer Mechanism is tasked with guiding the OS from entry to the Orientation Mechanism (see Fig. 12). One of the key components of the Transfer Mechanism subsystem is the Paddle Mechanism. The Paddle is attached to one side of the Transfer Mechanism. The diameter of the COM volume changes with respect to the central axis. Therefore, the paddle must change its diameter to avoid the OS jamming or getting pinned along the sides of the COM volume.

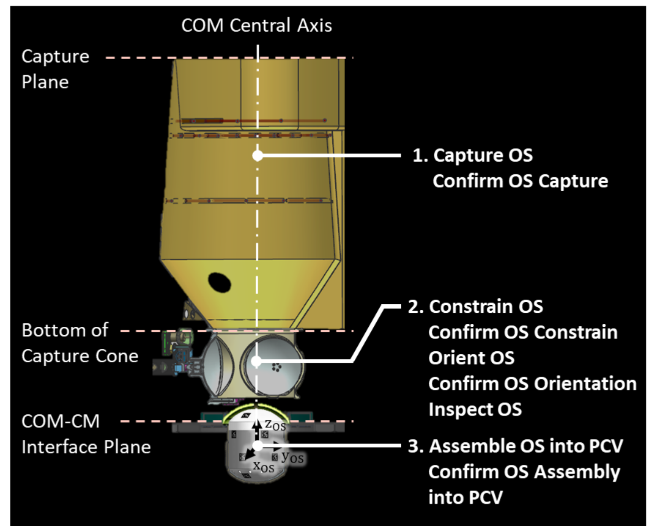

Figure 12. COM architecture concept definitions.

If the COM engineering team wanted to take a look at the various perspectives the Paddle is involved in, the model overview diagram is the place to start. The Paddle is integrated into every aspect of the model, as it plays a role in several key operational steps. The Paddle Mechanism is a part of the structural, behavioural, data, and requirements perspective.

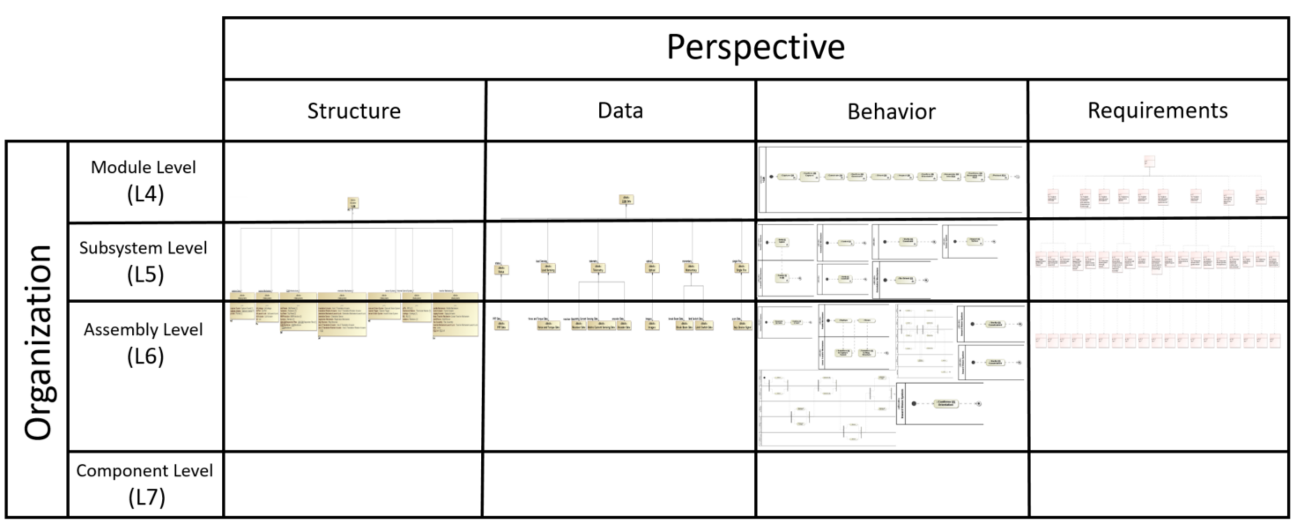

Figure 11. Full Capture and Orient Module model view.

Starting in the structure perspective's block definition diagram (BDD), the Paddle is listed as a part of the Transfer Mechanism subsystem (see Fig. 13).

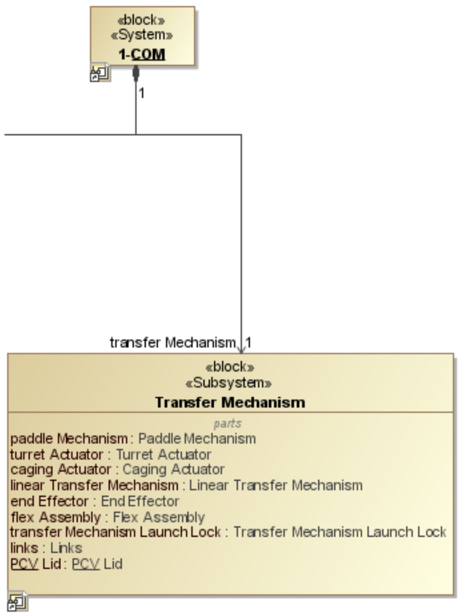

Figure 13. BDD of COM System broken down by Transfer Mechanism Subsystem.

This BDD then breaks down into several internal block diagrams (IBDs) for each subsystem listed. The IBD of the Transfer Mechanism shows: interactions between the Paddle and various other parts of the Transfer Mechanism, data flow, and power flow (see Fig. 14).

The IBD is exceedingly useful to the engineering team when defining various interface requirements as it visually documents all of the subsystem parts in one place, and clearly shows the flow of information. IBDs also help to show the flow of data and power between components. During the design phase of the Paddle, various space, repeatability, and reliability constraints provoked the engineering design team to switch from a passive paddle to an actively controlled paddle. Once an actuator was added to the Paddle Mechanism design, it was crucial to capture the various data and power items entailed. Since the Paddle would be operating in a space environment, the motor needs to remain at an operational temperature, thus heat must be provided from the COM system to the Paddle Actuator to keep it functional. The COM system needs to be cognizant of the position of the Paddle at any given time during capture and orientation operations, therefore encoder and limit switch data must be shared with the system.

Once the various data types that are generated and used by the system are determined, they can be organized categorically into a data hierarchy diagram. This hierarchy (see Fig. 15) is also placed on the overall model diagram, so

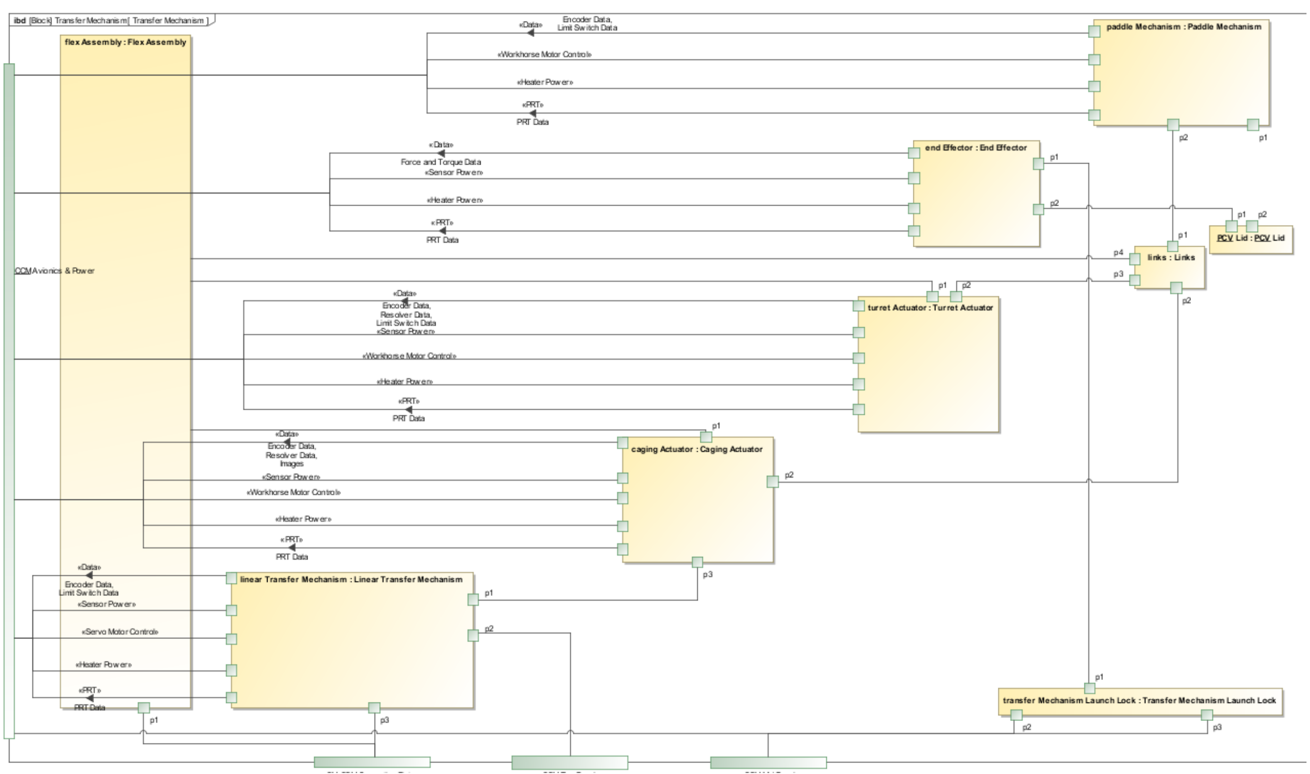

Figure 14. Transfer Mechanism IBD.

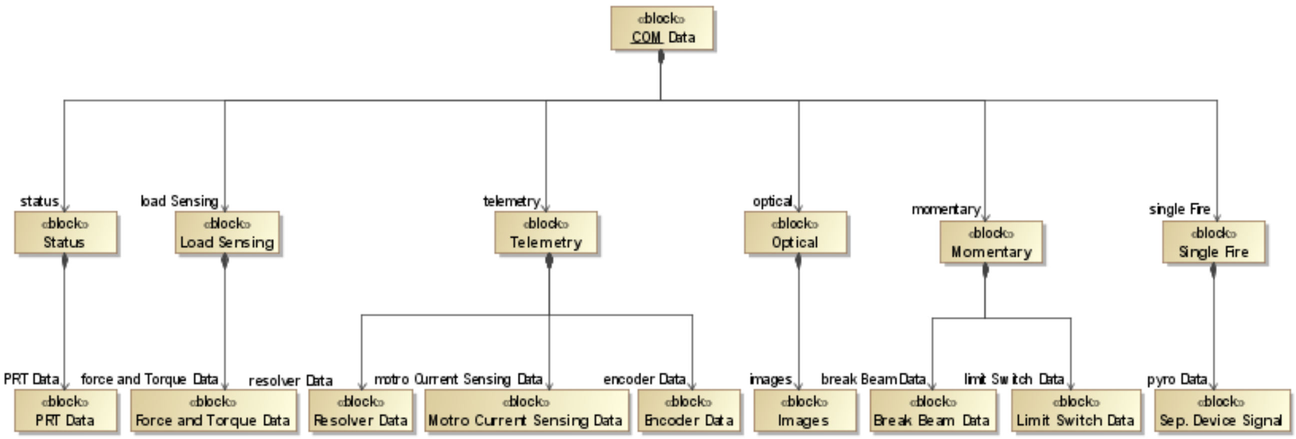

Figure 15. Hierarchy Diagram of Data Types collected by the COM system and subsystems.

that it may be easily referenced by the COM engineering team.

During capture operations, the Paddle's goal is to constrain the OS as soon as it is detected within the COM. The Transfer Mechanism would rotate from its stowed position into the COM central axis where the Paddle would expand to act as redundancy for the Capture Mechanism Lid. Activity diagrams, such as Fig. 16, are used to capture the Paddle's behaviour.

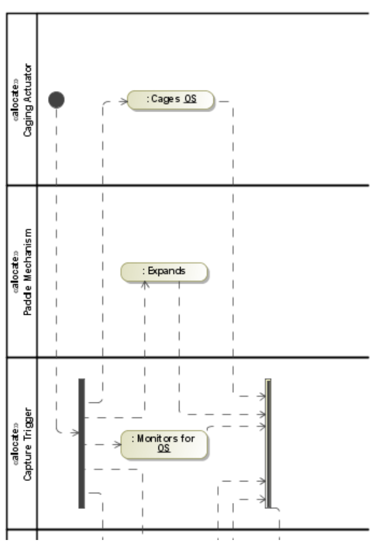

Figure 16. Activity diagram of OS Capture.

Activity diagrams show the flow of operations that take place within a system. In these diagrams, each swimlane designates a system or subsystem in which the activities are taking place. The Paddle is active during numerous sections of the COM operations, therefore activity diagrams such as Fig. 16 are extremely useful in communicating the complexity of the operational flow.

Furthermore, system requirements that pertain to the Paddle Mechanism can be located on the requirements block diagram within the overall systems model. These requirements are easily traced to the Paddle Mechanism, and can be referenced during COM engineering team meetings. As requirements change or are updated, the model acts as a resource that the COM engineering team members can use as a reference while working. Requirements can be presented in many diagrams and tables, however a hierarchal diagram clearly shows the parent/child relationships between requirements. Fig. 17 shows the breakdown from COM system requirements, to Transfer Mechanism subsystem requirements, finally to the Paddle component requirements.

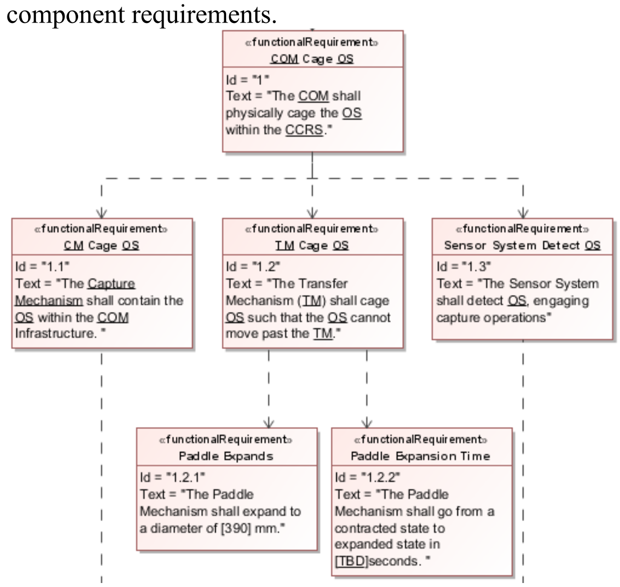

Figure 17. Caging requirement breakdown, including Transfer and Paddle Mechanism requirements.

5. Discussion

MBSE is a powerful method of system and project design that encourages thoughtful communication and insightful thinking amongst engineering teams. When implemented properly, MBSE tools such as SysML can produce models that become the central source of information that engineers within a team can look to and find the information most pertinent to them and their components. When the model becomes the only source engineers need to turn to when they have a question regarding the system, efficiency within the team is improved. Productivity is increased when instead of looking for information that one person has on one document on their computer, the engineer can simply open the system model and find whatever information they are in need of.

The model also has the potential to be an excellent trigger of thoughtful conversations and discussions regarding the system design, as complex designs and ideas can be neatly presented to the whole engineering team. MBSE provokes engineers to think about interfaces early on in the design process, which increases the conversation between people working on different subsystems and components.

6. Conclusion and Future Work

A model-based systems engineering (MBSE) approach was applied to architecting an orbiting sample Capture and Orient Module (COM) system concept for a Capture, Contain, and Return System (CCRS) payload concept for potential Mars Sample Return (MSR) at the NASA Jet Propulsion Laboratory. An architecture framework was established, covering multiple organizational layers of the system, along with structural, behavioral, data, and requirements perspectives. A workflow process to implement the architecting activities within the COM engineering team was developed.

Future development of the COM system model will include the implementation of use case diagrams to construct power and data profiles for various mission scenarios. Use case diagrams will also help the COM engineering team to better understand the various implications of design and operational decisions. Further expansion of system, subsystem, and component requirements are necessary to the completeness of the COM system model.

Acknowledgements

The research described in this publication was carried out at the Jet Propulsion Laboratory of California Institute of Technology under contract from the National Aeronautics and Space Administration (NASA). The decision to implement Mars Sample Return will not be finalized until NASA's completion of the National Environmental Policy Act (NEPA) process. This document is being made available for information purposes only.

References

[1] Younse, P., C. Chiu, J. Cameron, M. Dolci, E. Elliot, A. Ishigo, D. Kogan, E. Marteau, V. Malyan, J. Mayo, J. Munger, P. Ohta, E. Olds, L. Strahle, and R. Willson, "Concept for an On-orbit Capture and Orientation Module for Potential Mars Sample Return," *IEEE Aerospace Conference*, Big Sky, MT, Mar. 7-14, 2020.

[2] Way, D., "Preliminary Assessment of the Mars Science Laboratory Entry, Descent, and Landing Simulation," *IEEE Aerospace Conference*, Big Sky, MT, Mar. 2-9, 2013.

[3] Grotzinger, J. P., J. Crisp, A. R. Vasavada, R. C. Anderson, C. J. Baker, R. Barry, D. F. Blake, P. Conrad, K. S. Edgett, B. Ferdowski, R. Gellert, J. B. Gilbert, M. Golombek, J. Gómez-Elvira, D. M. Hassler, L. Jandura, M. Litvak, P. Mahaffy, J. Maki, M. Meyer, M. C. Malin, I. Mitrofanov, J. J. Simmonds, D. Vaniman, R. V. Welch, and R. C. Wiens, "Mars Science Laboratory Mission and Science Investigation," *Space Sci. Rev.*, 170:5-56, 2012.

[4] Guizzo, E., "Planetary Rovers: Are We Alone?," *IEEE Spectrum*, Dec. 30, 2010. [Online]. Available: https://spectrum.ieee.org/aerospace/robotic-exploration/planetary-rovers-are-we-alone. [Accessed Sept. 18, 2019].

[5] Smith, A., "How Many Jobs did the Mars Landing Create?," *CNN Money*, Aug. 6, 2012. [Online]. Available: https://money.cnn.com/2012/08/06/news/companies/mars-curiosity-jobs/index.htm. [Accessed Sept. 18, 2019].

[6] United States Government Accountability Office, *NASA Assessments of Major Projects*, GAO-12-207SP NASA, Mar. 2012.

[7] United States Government Accountability Office, *NASA Assessments of Selected Large-Scale Project*, GAO-09-306SP NASA, Mar. 2009.

[8] Welch, R., D. Limonadi, and R. Manning, "Systems Engineering the Curiosity Rover: A Retrospective," *2013 8th International Conference on System of Systems Engineering*, 70-75, 2013.

[9] Mustard, J.F., M. Adler, A. Allwood, D.S. Bass, D.W. Beaty, J.F. Bell III, W.B. Brinckerhoff, M. Carr, D.J. Des Marais, B. Drake, K.S. Edgett, J. Eigenbrode, L.T. Elkins-Tanton, J.A. Grant, S. M. Milkovich, D. Ming, C. Moore, S. Murchie, T.C. Onstott, S.W. Ruff, M.A. Sephton, A. Steele, and A. Treiman, *Report of the Mars 2020 Science Definition Team*, 154 pp., posted July, 2013, by the Mars Exploration Program Analysis Group (MEPAG), 2013.

[10] Williford, K., K. Farley, K. Stack, A. Allwood, D. Beaty, L. Beegle, R. Bhartia, A. Brown, M. de la Torre Juarez, S. Hamran, M. Hecht, J. Hurowitz, J. Rodriguez-Manfredi, S. Maurice, S. Milkovich, R. Wiens, Roger, "The NASA Mars 2020 Rover Mission and the Search for Extraterrestrial Life," *Habitability to Life on Mars,* 275-308, 2018.

[11] National Aeronautics and Space Administration, *FY2020 Explore Budget Estimates*, NASA, 2019.

[12] Witze, A., "The $2.4-billion Plan to Steal a Rock from Mars," *Nature*, Jan. 18, 2017. [Online]. Available: https://www.nature.com/news/the-2-4-billion-plan-to-steal-a-rock-from-mars-1.21306. [Accessed Oct. 9, 2019].

[13] United States Government Accountability Office, *NASA Assessments of Major Projects*, GAO-19-262SP NASA, May 2019.

[14] United States Government Accountability Office, *NASA Assessments of Major Projects*, GAO-17-303SP NASA, May 2017.

[15] Dvorak, D., "NASA Study on Flight Software Complexity," NASA Office of Chief Engineer, 2009.

[16] Butler, R. and M. Pennotti, "The Evolution of Software and Its Impact on Complex System Design in Robotic Spacecraft Embedded Systems," *2013 Conference on Systems Engineering Research*, Atlanta, GA, Mar. 19-22, 2013.

[17] *JPL 2018 Strategic Implementation Plan*, Jet Propulsion Laboratory, California Institute of Technology, 2018.

[18] "Joint Statement of Intent between the National Aeronautics and Space Administration and the European Space Agency on Mars Sample Return," NASA, ESA, 2018.

[19] National Academies of Sciences, Engineering, and Medicine, *Visions into Voyages for Planetary Sciences in the Decade 2013-2022: A Midterm Review*, Washington, DC: The National Academies Press, 2018.

[20] Muirhead, B. and A. Karp, "Mars Sample Return Lander Mission Concepts," *2019 IEEE Aerospace Conference*, Big Sky, MT, Mar. 2-9, 2019.

[21] Topper, J. and N. Horner, "Model-Based Systems Engineering in Support of Complex Systems Development," *John Hopkins APL Technical Digest*, v. 32, no. 1, 2013.

[22] Carrol, E. and R. Malins, *Systematic Literature Review: How is Model-Based Systems Engineering Justified?* Sandia National Laboratories, 2016.

[23] Friedenthal, S., A. Moore, and R. Steiner, *A Practical Guide to SysML,* 3rd ed. Waltham, MA: Elsevier, 2015.

[24] Madni, A. and S. Purohit, "Economic Analysis of Model-Based Systems Engineering," *Systems*, 7, 12, 2019.

[25] Chhaniyara, S., C. Saaj, M. Althoff-Kotzias, I. Ahrns, and B. Maediger, "Model Based Systems Engineering for Space Robotics Systems," *11th Symposium on Advanced Space Technologies in Robotics and Automation*, ESTEC, Noordwijk, The Netherlands, Apr. 12-14, 2011.

[26] Rahman, M. and M. Mizukawa, "Model-Based Development and Simulation for Robotic Systems with SysML, Simulink and Simscape Profiles," *International Journal of Advanced Robotic Systems*, vol. 10, 112, 2013.

[27] Elaasar, M., N. Rouquette, S. Jenkins, and S. Gerard, "The Case for Integrated Model Centric Engineering," *Proceedings of Model Based Enterprise Summit*, Gaithersburg, MD, Apr. 2019.

[28] Haskins, C., "A Historical Perspective of MBSE with a View to the Future," *INCOSE International Symposium*, 21(1), 2011.

[29] Kim, S., D. Wagner, and A. Jimenez, "Challenges in Applying Model-Based Systems Engineering: Human-Centered Design Perspective," *INCOSE Human Systems Integration Conference*, Biarritz, France, Sept. 11-13, 2019.

[30] Huckaby, J. and H. Christensen, "A Case for SysML in Robotics," *2014 IEEE Conference on Automation Science and Engineering,* Taipei, Taiwan, Aug. 18-22, 2014.

[31] Kornfeld, R., J. Parrish, S. Sell, and S. May, "Mars Sample Return: Testing the Last Meter of Rendezvous and Sample Capture," *Journal of Spacecraft and Rockets*, 44(3), Jun. 2007.

[32] Mukherjee, R., B. Chamberlain-Simon, R. Smith, M. Dolci, R. McCormick, and P. Ohta, "Concepts for Mars On-Orbit Robotic Sample Capture and Transfer," *IEEE Aerospace Conference*, Big Sky, MT, Mar. 4-11, 2017.

[33] Parrish, J., "To Mars and Back: Technologies for a Potential NASA Mars Sample Return," Mountain View, CA, Jun. 14, 2017.

[34] Younse, P., J. Strahle, M. Dolci, P. Ohta, K. Lalla, and E. Olds, "An Orbiting Sample Capture and Orientation Element Architecture for Potential Mars Sample Return," *2018 IEEE Aerospace Conference*, Big Sky, MT, Mar. 3-10, 2018.

[35] Younse, P., R. Adajian, B. Cano, M. Dolci, P. Ohta, K. Lalla, V. Malyan, J. Munger, E. Olds, and J. Strahle, "Sample Capture and Orientation Technologies for Potential Mars Sample Return," *Proc. 2nd International Mars Sample Return 2018*, 6027, Apr. 25, 2018.

[36] National Aeronautics and Space Administration, *NASA Technology Roadmaps, Introduction, Crosscutting Technologies, and Index*, Washington, DC: Author, 2015.

[37] Borky, J. and T. Bradley, *Effective Model-Based Systems Engineering*, Cham: Switzerland: Springer, 2018.

[38] National Aeronautics and Space Administration, *NASA Systems Engineering Handbook*, NASA SP-2016-6105 Rev2, Washington, DC: Author, 2016.

[39] National Aeronautics and Space Administration, *NASA Systems Engineering Processes and Requirements*, NASA Procedural Requirement 7123.1B, Washington, DC: Author, 2013.

[40] Aleksandraviciene, A. and A. Morkevicius, *MagicGrid Book of Knowledge*, Kaunas, Lithuania: Vitae Litera, 2018.

[41] Pohl, K., *Requirements Engineering*, Berlin, Germany: Springer, 2010.

BIOGRAPHY

Paulo Younse *is a robotics engineer for the Robotic Systems Group at NASA's Jet Propulsion Laboratory in Pasadena, CA. His experience resides in mechanical design, machine vision, systems engineering, systems architecting, and planetary sample collection. Previous experience includes development of the Mars 2020 rover sample caching architecture and sample tube hermetic seals to preserve samples for potential Earth return, work on unmanned underwater vehicles at the Boeing Company, and visual navigation and control for agricultural robots at the University of Florida. He received a B.S. in Mechanical Engineering from California Polytechnic State University, San Luis Obispo, and an M.E. in Agricultural Engineering from the University of Florida.*

Jessica Cameron *is an undergraduate student at California State Polytechnic University, Pomona, currently working towards a B.S. in Aerospace Engineering. Previously, she worked at NASA's Jet Propulsion Laboratory executing various tests and prototyping for Axel and DuAxel robots. Currently, she is thrilled to be assisting with Systems Engineering and Program Management for the Mars Sample Return Campaign.*